

In the Claims

This listing of claims will replace all prior versions and listings of claims in the application:

- 1 1. (Currently Amended) A pipelined data processor operating
2 in a plurality of pipeline phases including at least an instruction
3 decode pipeline phase and an execution pipeline phase capable of
4 predicated instruction execution dependent upon the state of an
5 instruction designated predicate register comprising:
 - 6 a data register file including a plurality of read/write,
7 general purpose data registers;
 - 8 an instruction decode unit operative during ~~an said~~ instruction decode pipeline phase receiving fetched instructions
9 and determining the identity of at least one source operand data
10 register, a destination operand data register and one of a plurality
11 of functional units for execution of each instruction,
12 said instruction decode unit further identifying a predicate
13 register responsive to receipt of a predicated instruction;
14 a scoreboard bit corresponding to each data register capable
15 of serving as a predicate register, each scoreboard bit connected
16 to said instruction decode unit to be set to a first digital state
17 upon determining said corresponding data register is a destination
18 for an instruction and connected to said plurality of functional
19 units to be reset to a second digital state opposite to said first
20 digital state upon functional unit write of a result to said
21 corresponding data register;
 - 22 said plurality of functional units operative during an execution pipeline phase connected to said instruction decode unit for performing a data processing operation on at least one source operand recalled from at least one corresponding instruction designated source data register and producing a result, said functional unit

29 responsive to an instruction not a predicate instruction
30 to write said result to an instruction designated destination
31 data register, and

32 responsive to a predicate instruction to write said
33 result to an instruction designated destination data register
34 if said corresponding predicate data register has a first
35 state during said execution pipeline phase regardless of said
36 state of said corresponding scoreboard bit and to nullify said
37 instruction and not write said result if said predicate
38 register has a second state opposite to said first state
39 during said execution pipeline phase regardless of said state
40 of said corresponding scoreboard bit; and

41 a scoreboard bit corresponding to each data register capable
42 of serving as a predicate register, each scoreboard bit connected
43 to said instruction decode unit to be set to a first digital state
44 upon determining said corresponding data register is a destination
45 for an instruction and connected to said plurality of functional
46 units to be reset to a second digital state opposite to said first
47 digital state upon functional unit write of a result to said
48 corresponding data register, and

49 each functional unit is further operative responsive to a
50 predicate instruction during said instruction decode pipeline phase
51 to nullify said predicate instruction of a following execution
52 phase by operating at a reduced power state relative to normal
53 instruction operation if said predicate register has said second
54 state during said instruction decode pipeline phase and said
55 corresponding scoreboard bit has said second state during said
56 instruction decode pipeline phase.

1 2. (Original) The pipelined data processor of claim 1,
2 wherein:

3 said functional unit is further operative to reset said
4 scoreboard bit to said second digital state upon nullification of
5 said instruction designating a corresponding data register as a
6 destination operand data register.

3. (Canceled)

1 4. (Currently Amended) A method of operating a pipelined
2 data processor operating in a plurality of pipeline phases
3 including at least an instruction decode pipeline phase and an
4 execution pipeline phase capable of predicated instruction
5 execution dependent upon the state of an instruction designated
6 predicate register comprising the steps of:

7 setting a scoreboard bit to a first digital state upon
8 determining a corresponding data register is a destination for an
9 instruction;

10 resetting a scoreboard bit to a second digital state opposite
11 to said first digital state upon a write of a result to said
12 corresponding data register;

13 performing a data processing operation via a corresponding
14 functional unit on at least one source operand recalled from at
15 least one corresponding instruction designated source data
16 register, producing a result and writing said result to an
17 instruction designated destination data register in response to an
18 instruction not a predicate instruction;

19 performing a data processing operation via a corresponding
20 functional unit on at least one source operand recalled from at
21 least one corresponding instruction designated source data register
22 and producing a result in response to a predicate instruction
23 designating a corresponding predicate data register and writing
24 said result to an instruction designated destination data register
25 regardless of said state of said corresponding scoreboard bit if

26 said corresponding predicate data register has a first state during
27 said execution pipeline phase;
28 nullifying a data processing operation of a predicate
29 instruction by not writing said result to the instruction
30 designated destination data register via said corresponding
31 functional unit regardless of said state of said corresponding
32 scoreboard bit if said corresponding predicate register has a
33 second state opposite to said first state during said execution
34 pipeline phase; and
35 nullifying a predicate instruction for a following execution
36 phase by operating said corresponding functional unit at a reduced
37 power state relative to normal instruction operation if said
38 corresponding predicate register has said second state during a
39 prior instruction decode pipeline phase and said corresponding
40 scoreboard bit has said second state during a prior instruction
41 decode pipeline phase.

1 5. (Original) The method of claim 4, further comprising the
2 step of:

3 resetting a scoreboard bit to a second digital state upon
4 nullification of said instruction designating said corresponding
5 data register as a destination operand data register.

6. (Canceled)

1 7. (Currently Amended) The method of claim 4 further
2 comprising the steps of:

3 statically scheduling instruction execution via a compiler;
4 and
5 scheduling via said compiler a last write to a data register
6 before a an instruction decode pipeline phase of a predicate
7 instruction designating said data register as a predicate register.

1 8. (Previously Presented) The pipelined data processor of
2 claim 1, wherein:
3 each functional unit is operable at said reduced power state
4 by not fetching at least one instruction operand and not toggling a
5 corresponding register read port during said following execution
6 phase.

1 9. (Currently Amended) ~~the~~ The pipelined data processor of
2 claim 1, wherein:
3 each functional unit is operable at said reduced power state
4 by not powering said functional unit during said following
5 execution phase.

1 10. (Previously Presented) The method of claim 4, wherein:
2 said step of operating said corresponding functional unit at a
3 reduced power state includes not fetching at least one instruction
4 operand and not toggling a correspond register read port during
5 said following execution phase.

1 11. (Previously Presented) The method of claim 4, wherein:
2 said step of operating said corresponding functional unit at a
3 reduced power state relative includes not powering said functional
4 unit during said following execution phase.